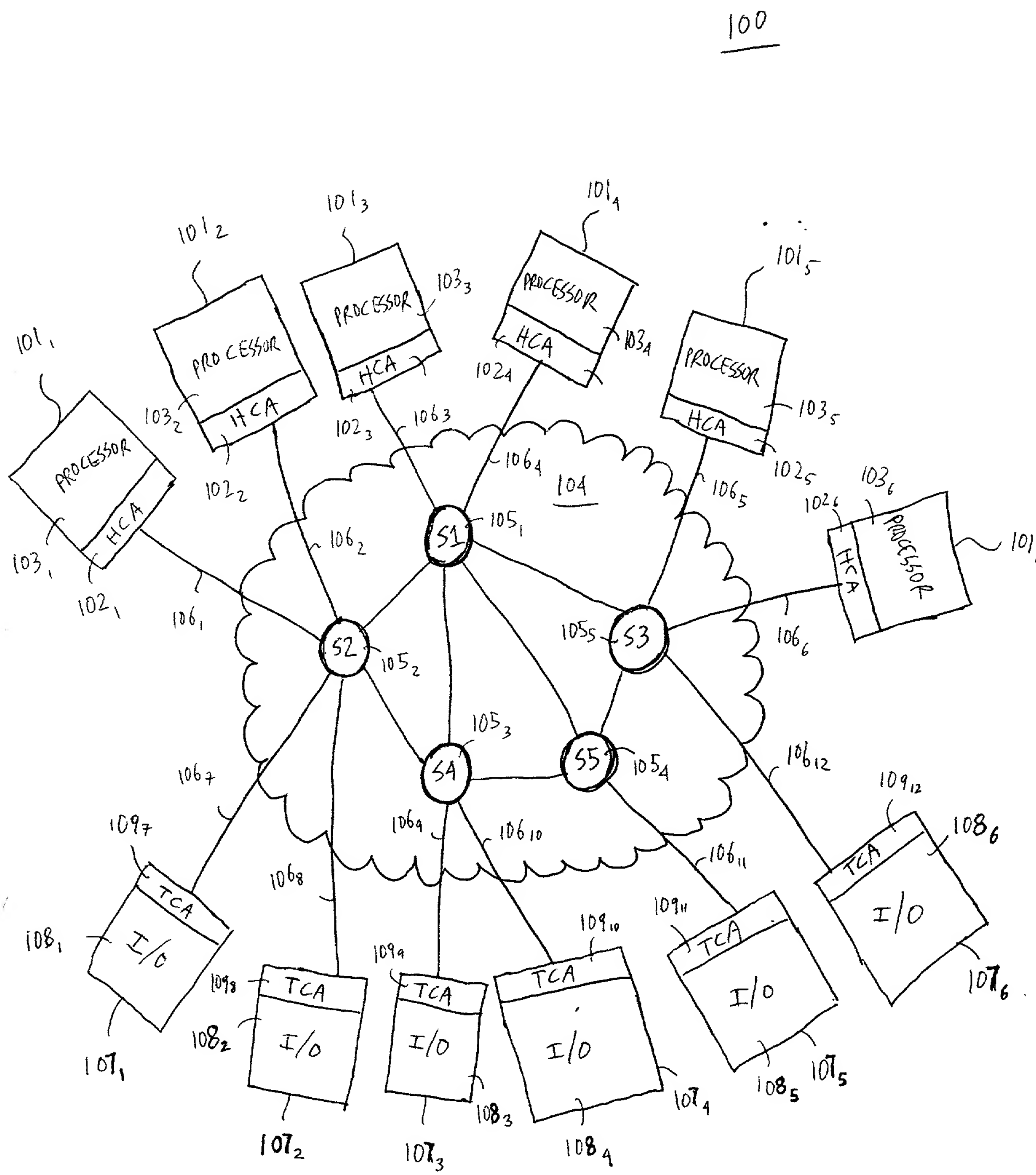
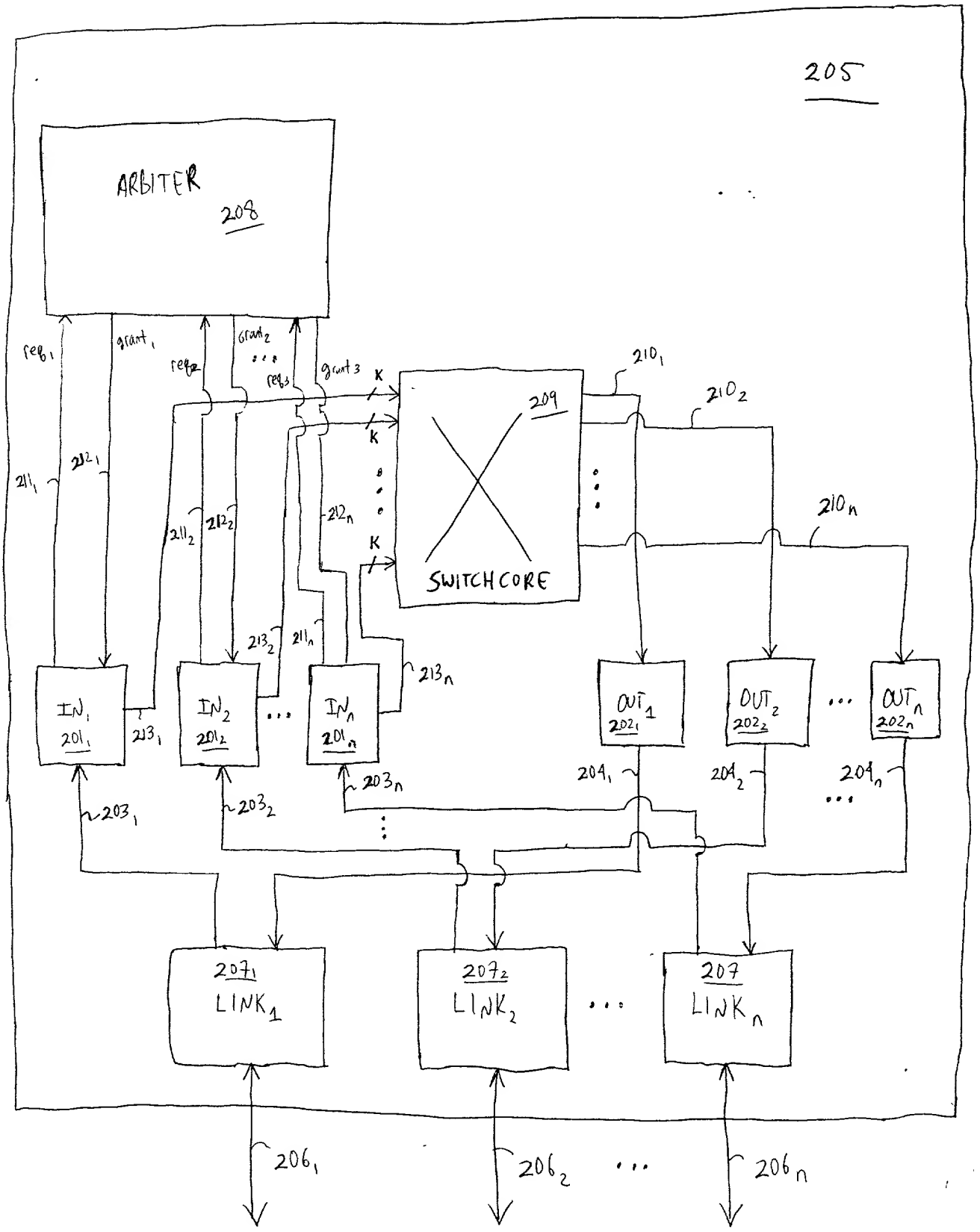


FIGURE 1

FIGURE 2


205
ARBITER
208
SWITCH CORE
209
req_1
$grant_1$
req_2
$grant_2$
req_3
$grant_3$
K
211_1
212_1
211_2
212_2
212_n
211_n
213_2
213_n
213_1
210_1
210_2
210_n
IN_1 201
IN_2 201
IN_n 201
OUT_1 202
OUT_2 202
OUT_n 202
203_1
203_2
203_n
204_1
204_2
204_n
207
$LINK_1$
$LINK_2$
$LINK_n$
206_1
206_2
206_n

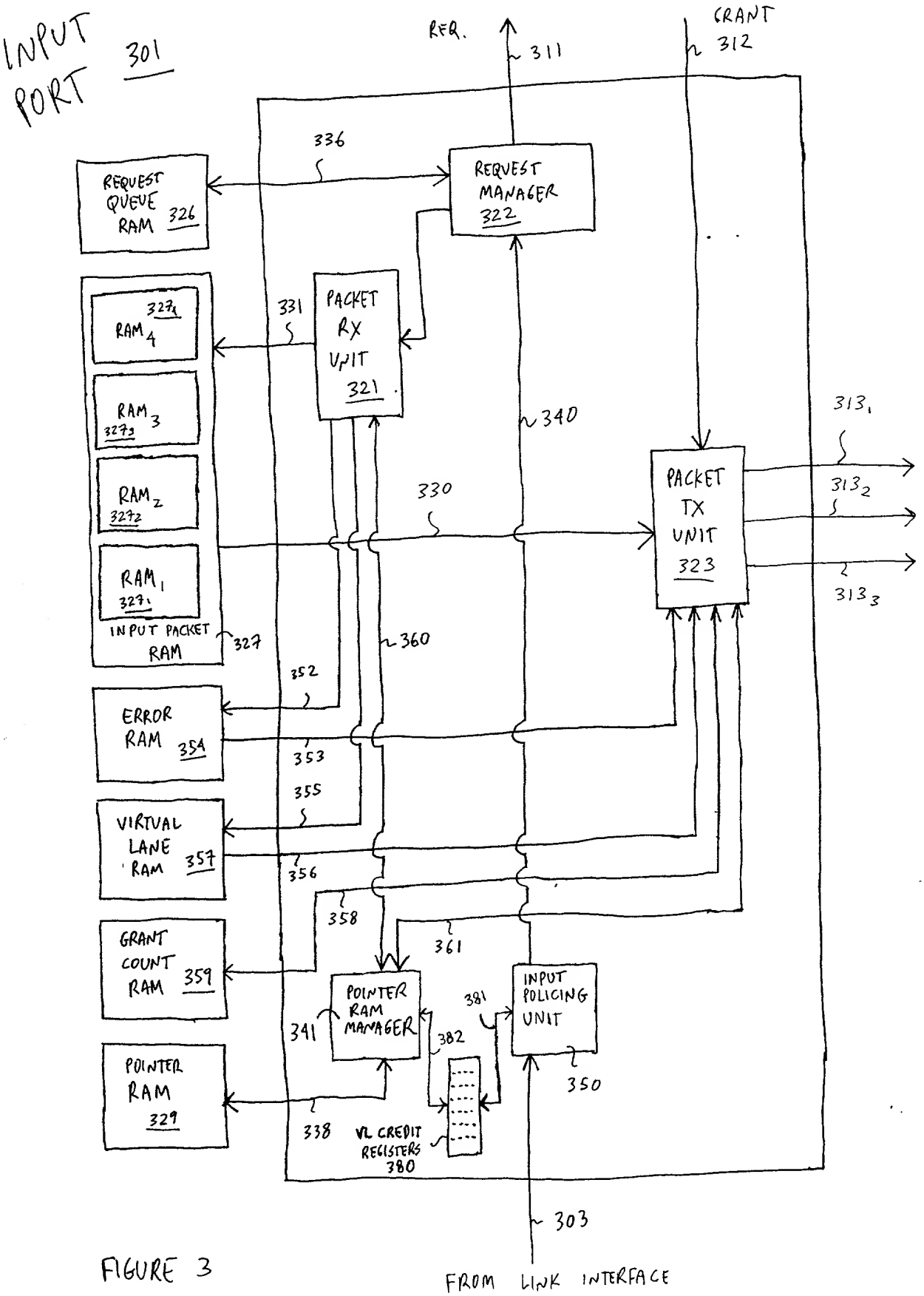

INPUT PORT 301
REQ. 311
GRANT 312
336
REQUEST QUEUE RAM 326
REQUEST MANAGER 322
PACKET RX UNIT 321
331
RAM 4 327 4
RAM 3 327 3
RAM 2 327 2
RAM 1 327 1
INPUT PACKET RAM 327
340
330
PACKET TX UNIT 323
313 1
313 2
313 3
360
352
ERROR RAM 354
353
355
VIRTUAL LANE RAM 357
356
358
361
GRANT COUNT RAM 359
POINTER RAM MANAGER
341
381
INPUT POLICING UNIT
382
350
POINTER RAM 329
338
VL CREDIT REGISTERS 380
303
FROM LINK INTERFACE

FIGURE 3

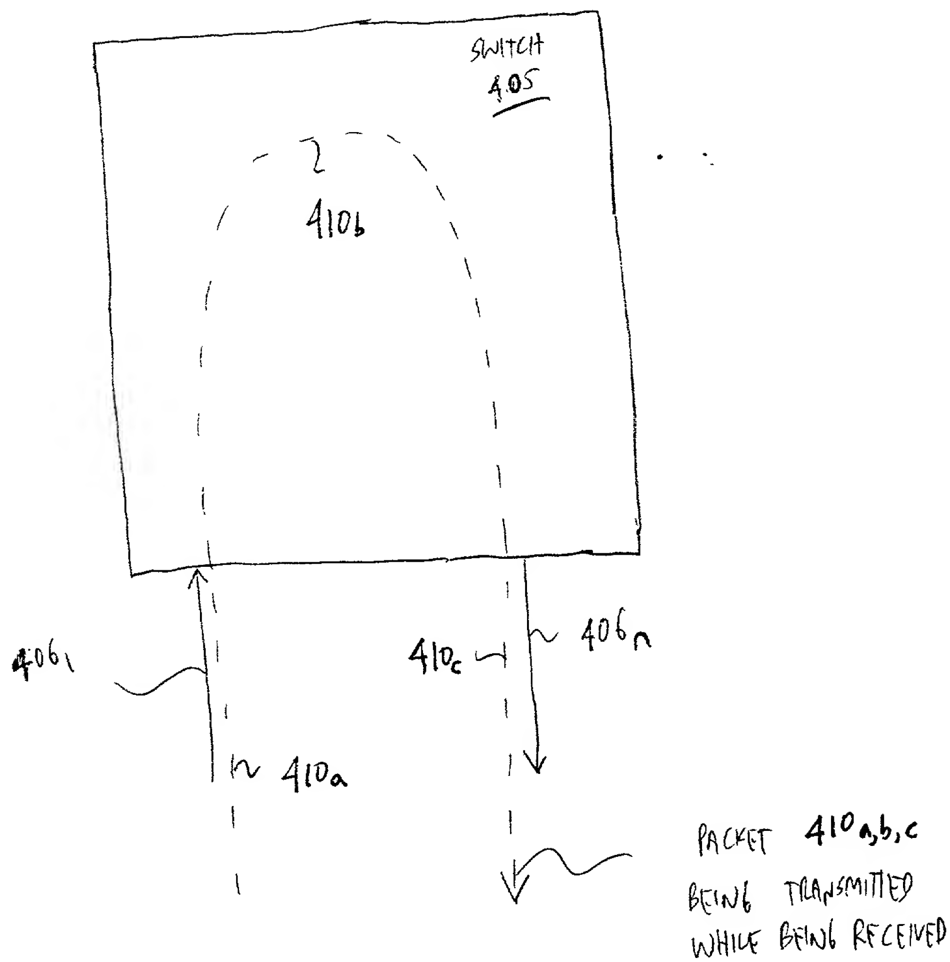

SWITCH
405
410b
406i
410c
406n
410a
PACKET 410a,b,c
BEING TRANSMITTED
WHILE BEING RECEIVED

FIGURE 4

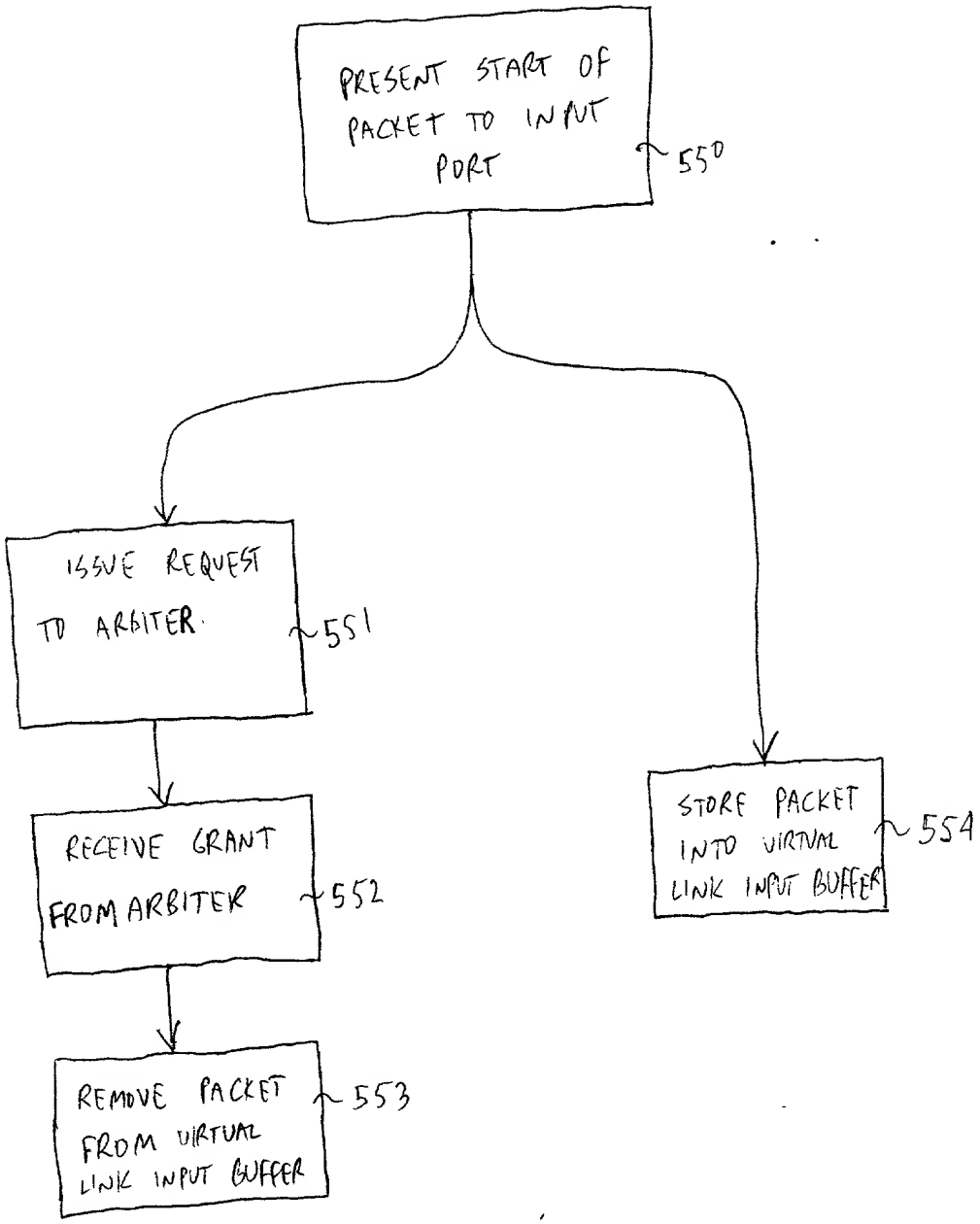

PRESENT START OF PACKET TO INPUT PORT
550
ISSUE REQUEST TO ARBITER.
551
RECEIVE GRANT FROM ARBITER
552
REMOVE PACKET FROM VIRTUAL LINK INPUT BUFFER
553
STORE PACKET INTO VIRTUAL LINK INPUT BUFFER
554

FIGURE 5

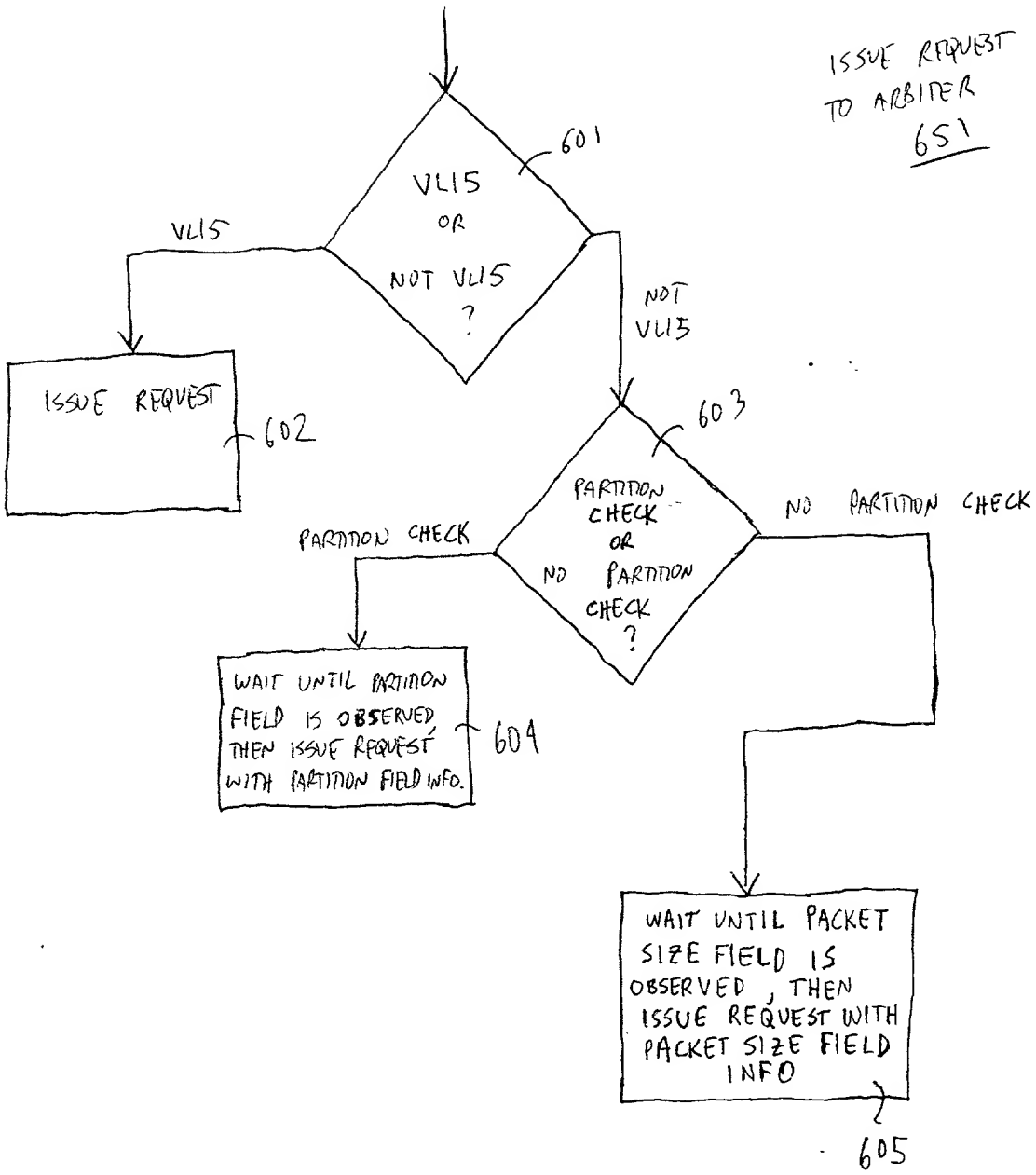

ISSUE REQUEST TO ARBITER 651
601
VL15 OR NOT VL15 ?
VL15
NOT VL15
ISSUE REQUEST
602
603
PARTITION CHECK OR NO PARTITION CHECK ?
PARTITION CHECK
NO PARTITION CHECK
WAIT UNTIL PARTITION FIELD IS OBSERVED, THEN ISSUE REQUEST WITH PARTITION FIELD INFO.
604
WAIT UNTIL PACKET SIZE FIELD IS OBSERVED, THEN ISSUE REQUEST WITH PACKET SIZE FIELD INFO
605

FIGURE 6

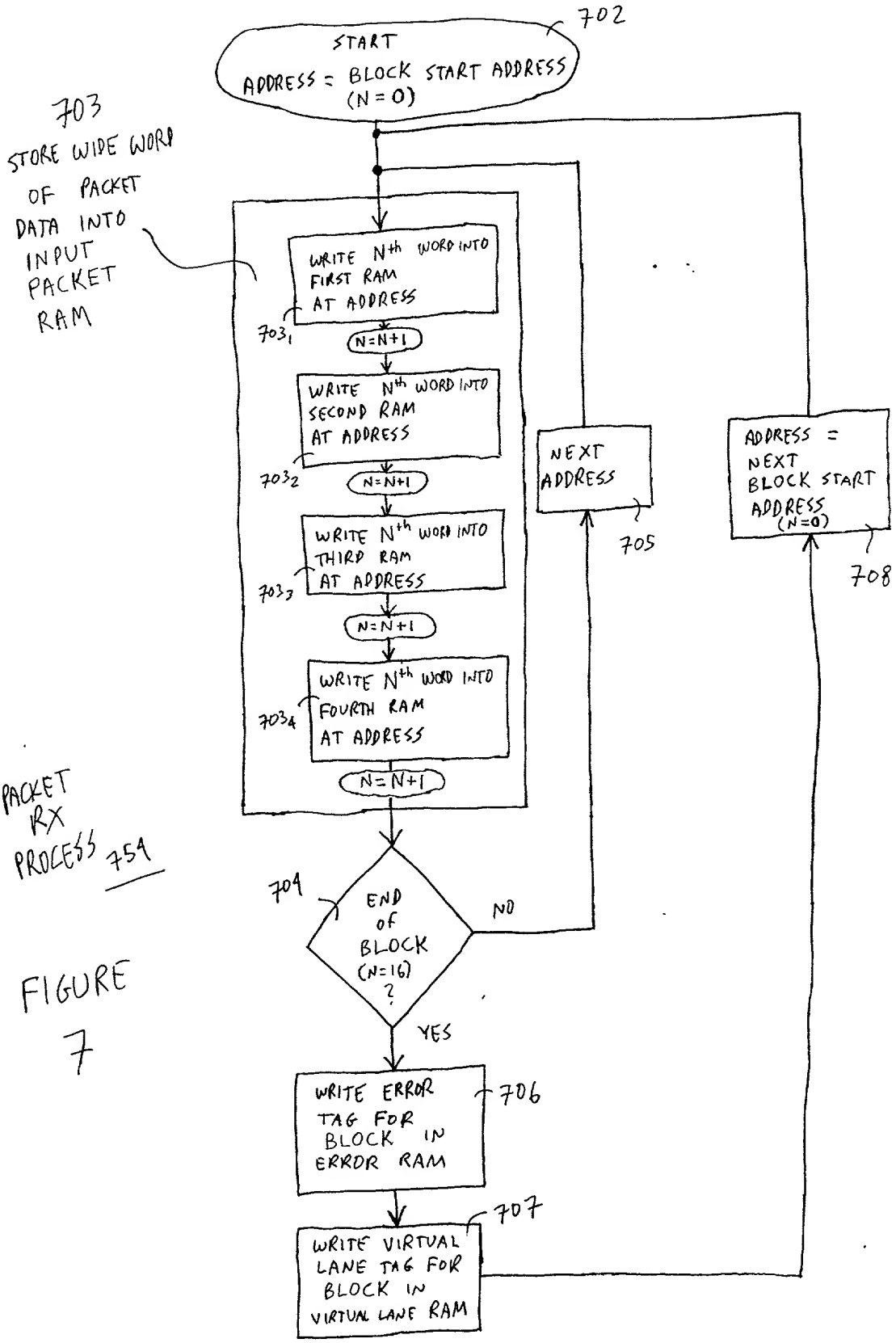

START
ADDRESS = BLOCK START ADDRESS
(N = 0)
702
703
STORE WIDE WORD OF PACKET DATA INTO INPUT PACKET RAM
WRITE Nth WORD INTO FIRST RAM AT ADDRESS
703₁
N=N+1
WRITE Nth WORD INTO SECOND RAM AT ADDRESS
703₂
N=N+1
WRITE Nth WORD INTO THIRD RAM AT ADDRESS
703₃
N=N+1
WRITE Nth WORD INTO FOURTH RAM AT ADDRESS
703₄
N=N+1
PACKET RX PROCESS 759
709
END OF BLOCK (N=16) ?
NO
YES
NEXT ADDRESS
705
ADDRESS = NEXT BLOCK START ADDRESS (N=0)
708
WRITE ERROR TAG FOR BLOCK IN ERROR RAM
706
WRITE VIRTUAL LANE TAG FOR BLOCK IN VIRTUAL LANE RAM
707

FIGURE 7

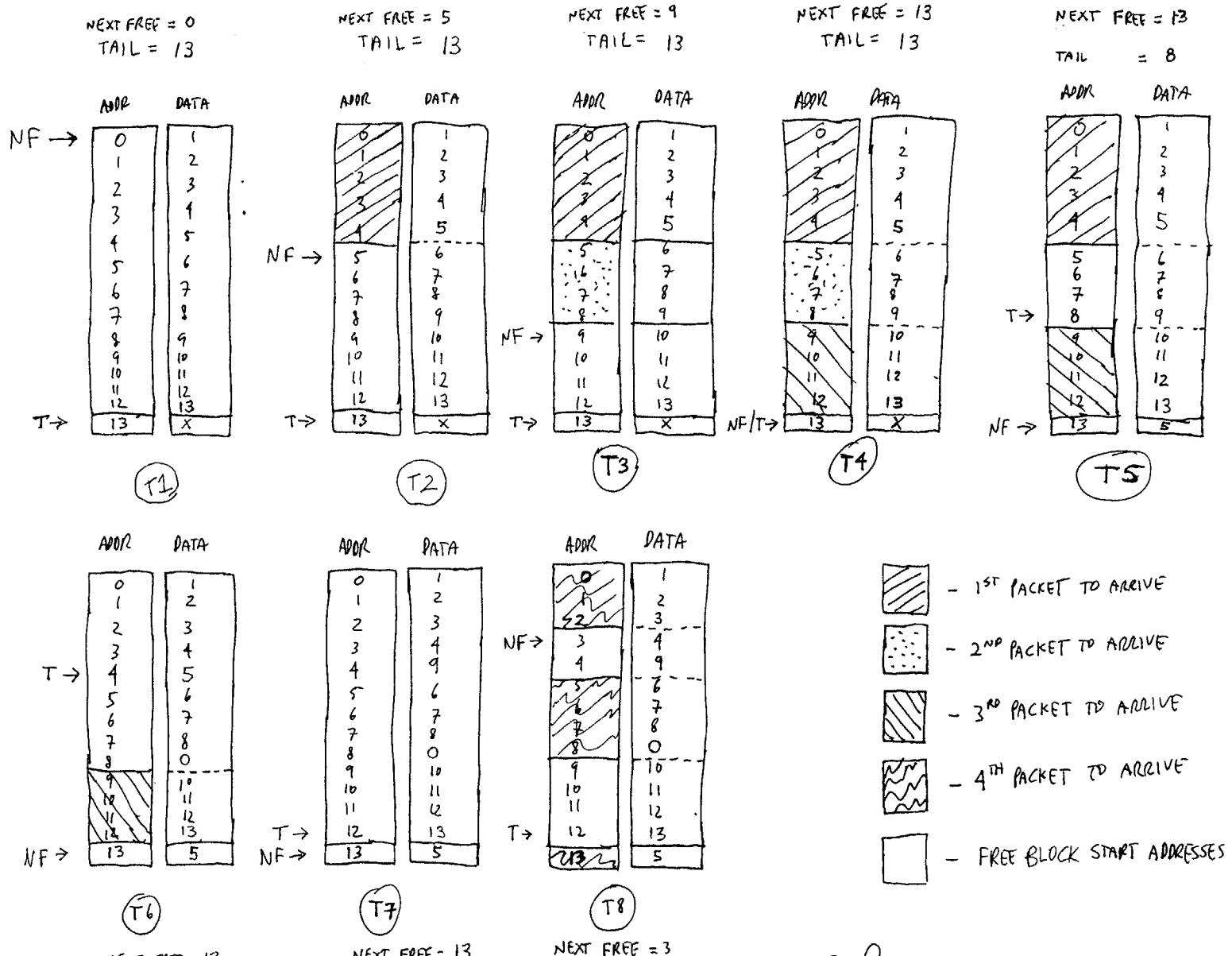

NEXT FREE = 0
TAIL = 13
ADDR
DATA
NF →
T →
T1
NEXT FREE = 5
TAIL = 13
T2
NEXT FREE = 9
TAIL = 13
T3
NEXT FREE = 13
TAIL = 13
NF/T →
T4
NEXT FREE = 13
TAIL = 8
T →
NF →
T5
T6
NEXT FREE = 13
TAIL = 4
T7
NEXT FREE = 13
TAIL = 12
T8
NEXT FREE = 3
TAIL = 12
- 1ST PACKET TO ARRIVE
- 2ND PACKET TO ARRIVE
- 3RD PACKET TO ARRIVE
- 4TH PACKET TO ARRIVE
- FREE BLOCK START ADDRESSES

FIGURE 8

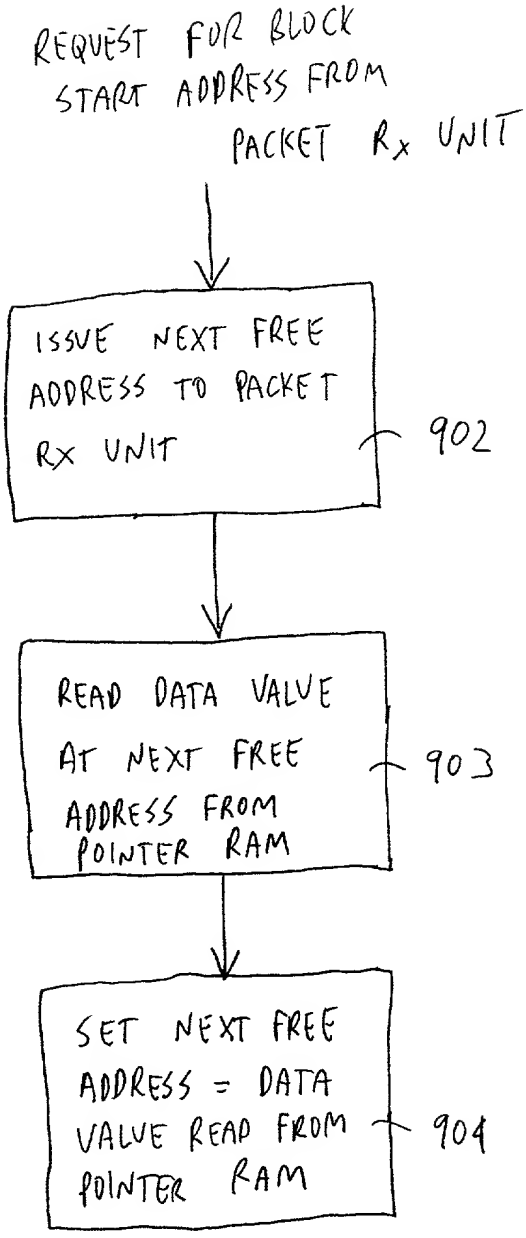

REQUEST FOR BLOCK START ADDRESS FROM PACKET Rx UNIT
ISSUE NEXT FREE ADDRESS TO PACKET Rx UNIT
902
READ DATA VALUE AT NEXT FREE ADDRESS FROM POINTER RAM
903
SET NEXT FREE ADDRESS = DATA VALUE READ FROM POINTER RAM
904

FIGURE 9A

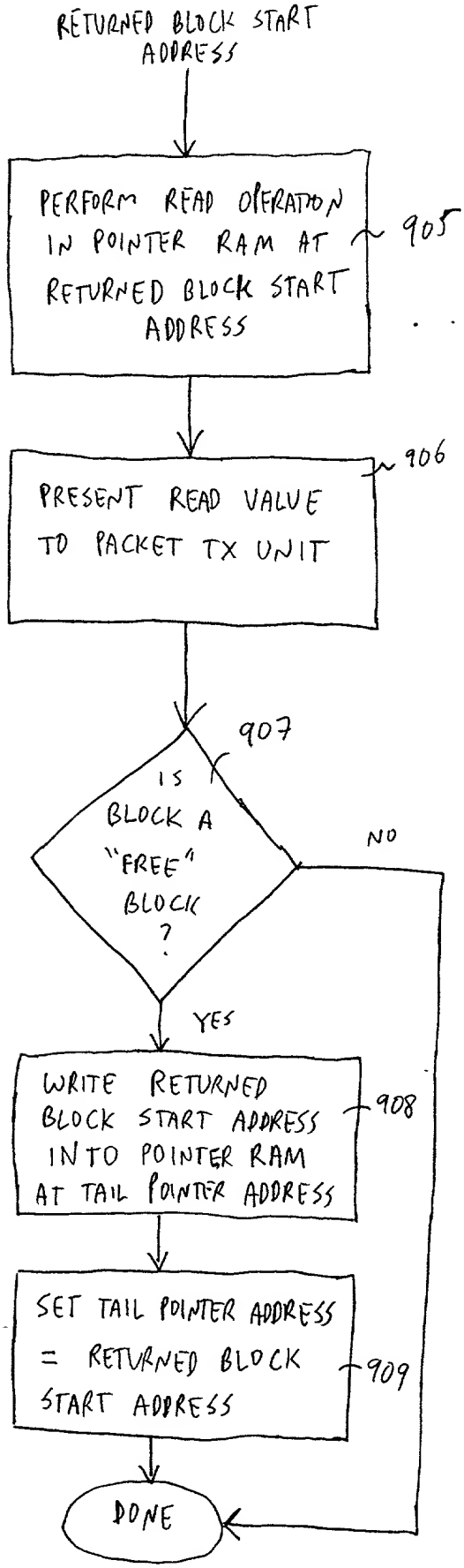
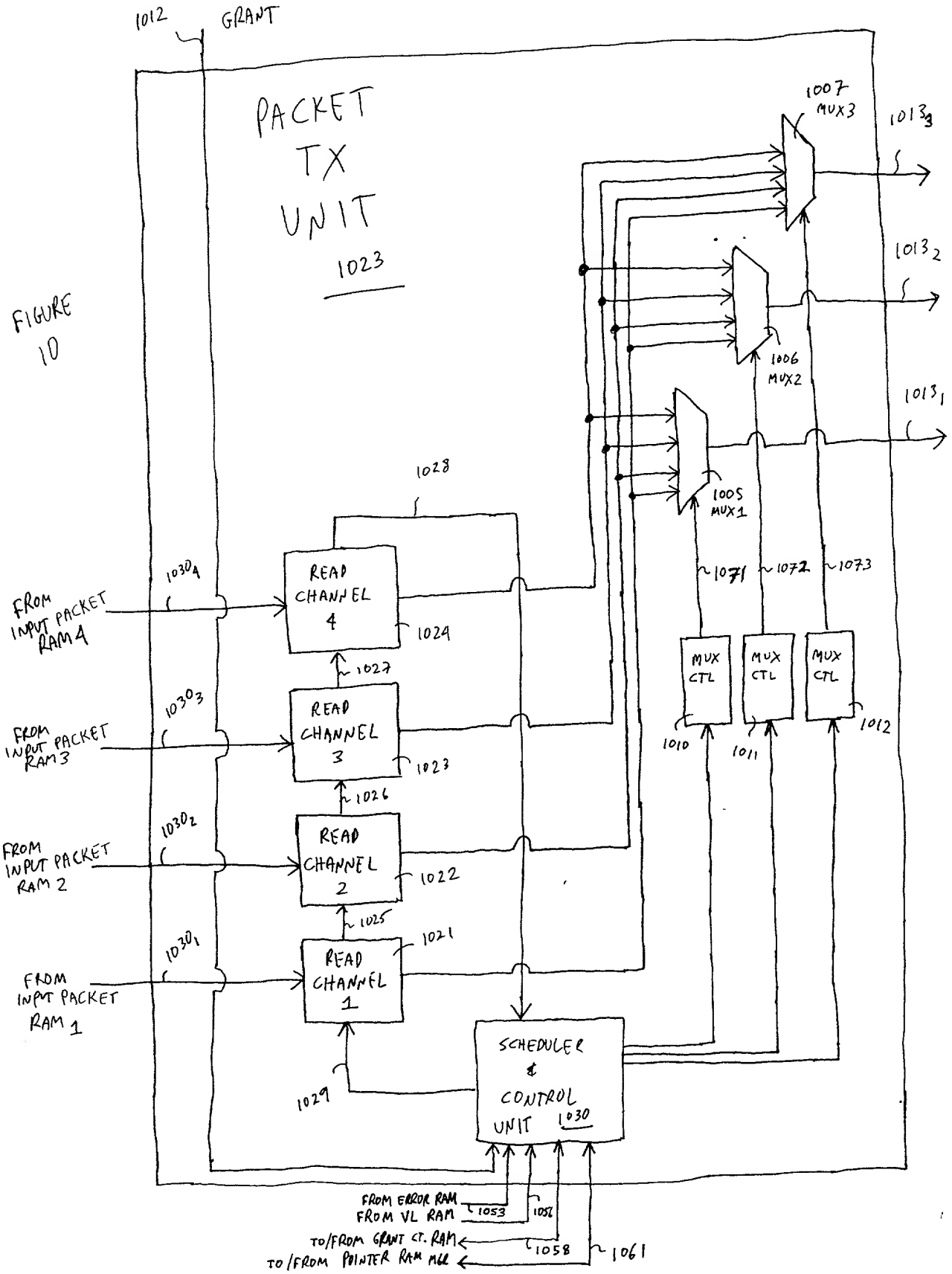

RETURNED BLOCK START ADDRESS
PERFORM READ OPERATION IN POINTER RAM AT RETURNED BLOCK START ADDRESS
905
PRESENT READ VALUE TO PACKET TX UNIT
906
907
IS BLOCK A "FREE" BLOCK ?
NO
YES
WRITE RETURNED BLOCK START ADDRESS INTO POINTER RAM AT TAIL POINTER ADDRESS
908
SET TAIL POINTER ADDRESS = RETURNED BLOCK START ADDRESS
909
DONE

FIGURE 9B

FIGURE 10



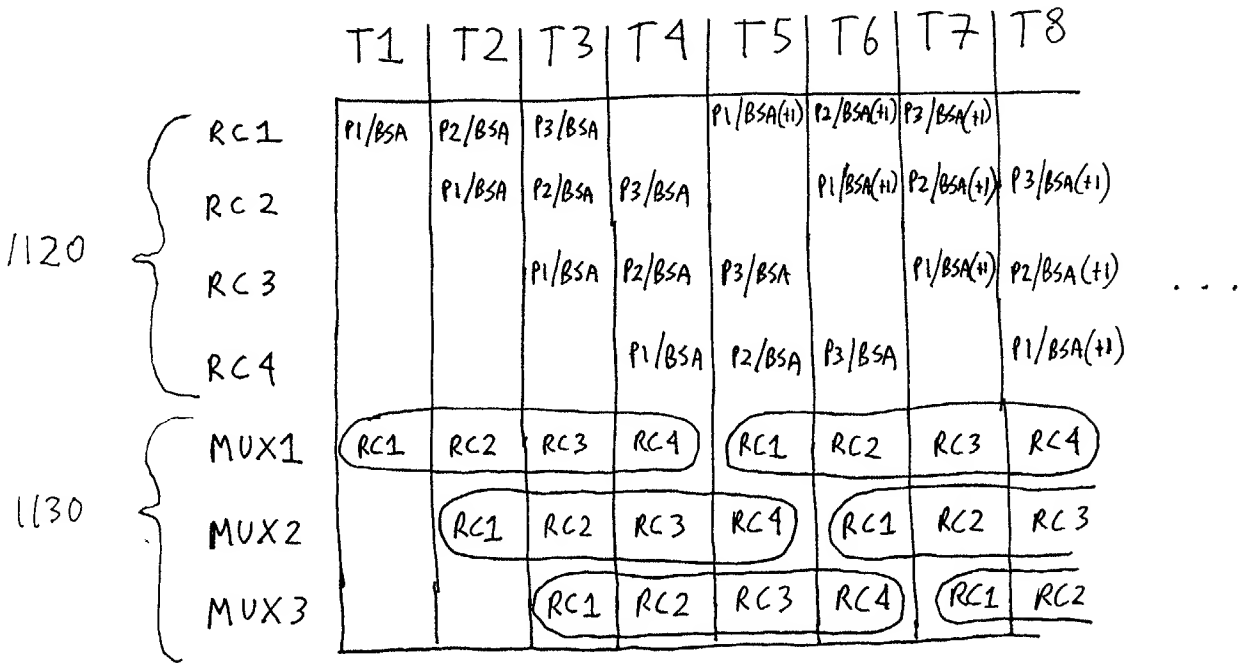

T1
T2
T3
T4
T5
T6
T7
T8
1120
RC1
RC2
RC3
RC4
P1/BSA
P2/BSA
P3/BSA
P1/BSA(+1)
P2/BSA(+1)
P3/BSA(+1)
1130
MUX1
MUX2
MUX3
RC1
RC2
RC3
RC4
Time

FIGURE 11